SNIA ■ SANTA CLARA, 2017

“Performance bugs” and How to Find Them

Rune Erlend Jensen

MemoScale / Norwegian University of Science and Technology

Per Simonsen

MemoScale

Abstract

- “Performance bugs” are performance anomalies.
- We identify memory address aliasing as an important mechanism causing performance bias on modern Intel microarchitectures.
- We link this effect to memory allocators and demonstrate that they can hurt performance in several cases.

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Presentation Overview

- Who are we?
- Optimization and data storage
- Optimization results
- Finding a "Performance Bug"
- Exploring the impacts

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Who are the presenters?

- Rune Erlend Jensen, Performance engineer
  - Extensive optimization experience
  - Ph.D. candidate at the Norwegian University of Science and Technology.
- Per Simonsen, CEO, MemoScale
  - MSc Entrepreneurship/Computer Science
  - MSc Business Administration

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

MemoScale

- Start-up with strong ties to the Norwegian University of Science and Technology.
- *We leverage our expertise in optimization, software engineering and coding theory to solve bottlenecks and improve performance in data storage systems.*

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Improving Performance

- Fastest unbroken SHA3 Candidate: Blue Midnight Wish
- Fastest Reed Solomon Erasure Coding implementation
- New erasure codes optimized for fast compute and rebuild

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Chasing Bottlenecks

- From hardware to software
- Hardware getting faster
- The CPU is still the center player

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Compute Intensive Components

- **Erasure coding**
- Hashing
- Encryption
- Compression

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Test Setup – Benchmark of MemoScale vs Jerasure and Intel ISA-L

Data blocks	10
Redundancy blocks	4
Block size	4 MB
Number of cores/threads	1
Measurement criteria	Data encoded excl. redundancies, MB/s
Codes tested	Reed Solomon and MemoScale compute optimized EC
Libraries tested	Jerasure, Intel ISA-L, MemoScale

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

EC Encoding Speed I – Intel Processor

Processor:
Intel Xeon
E5-2676 v3
2.4 Ghz

Erasure coding (EC):
* Reed Solomon

** MemoScale
compute optimized

MB/S
Data encoded

14000
12000
10000
8000
6000
4000
2000
0

Jerasure* | Intel ISA-L* | MemoScale* | MemoScale CO**

SDC 17

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

EC Encoding Speed II – ARM Processor

Processor:
HiSilicon Kirin 620 processor, ARM Cortex-A53 64-bit SoC 1.2ghz

Erasure coding (EC):
* Reed Solomon

** MemoScale compute optimized

MB/S
Data encoded

1200
1000
800
600
400
200
0

Jerasure*
MemoScale*
MemoScale CO**

SDC 17

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

EC Encoding Speed III – AMD Processor

Processor:
AMD RYZEN 7 1700 8-Core 3.0 GHz (3.7 GHz Turbo)

Erasure coding (EC):
* Reed Solomon

** MemoScale compute optimized

MB/S
Data encoded

16000
14000
12000
10000
8000
6000
4000
2000
0

Jerasure*
MemoScale*
MemoScale CO**

SDC 17

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Finding "Performance Bugs"

- What is a "Performance Bug"?
- Optimizing for performance

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Measurement Bias I

- What if changing your user name affected the execution time of programs?

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Measurement Bias II

- What if changing your user name affected the execution time of programs?
  - Runtime performance change from external factors, introducing bias.
    - Environment variables
    - Link ordering

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Measurement Bias III

- What if changing your user name affected the execution time of programs?
  - Changes in memory layout interacting with various hardware mechanisms.
    - Hard to predict :(

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Measurement Bias IV

- Challenge for performance analysis
  - Comparison of algorithms on different system configurations can give misleading results.
  - Effects can invalidate perceived speedup (Mytkowicz et al. [1]).

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Methodology I

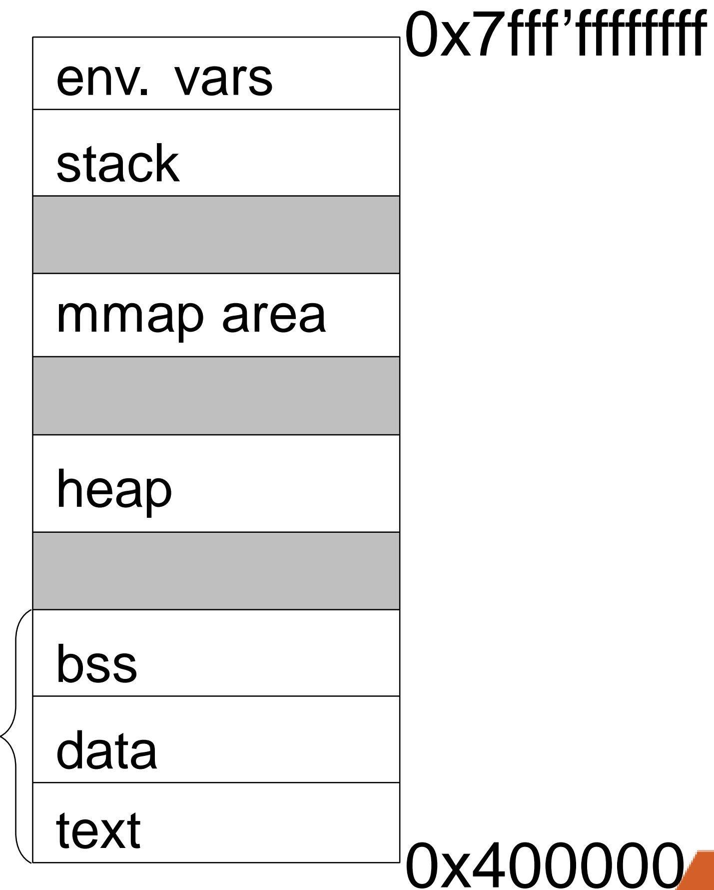
- Control execution
  - Control execution environment (disable ASLR*).
  - Instrument program using hardware performance counters.
  - Measure over range of environments.

*Address space layout randomization.

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Methodology II

- Memory execution context, assuming a 64-bit process running on a Linux system.



2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Bias from Environment Size I

- A simple program*, with no surprises?

```
static int i, j, k;

int main() { int g = 0, inc = 1;
   for (; g < 65536; g++) { i +=
   inc; j += inc; k += inc;
   }
   return 0;
}
```

*Microkernel first presented by Mytkowicz *et al.*[2], showing env. bias.

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Bias from Environment Size II

- Execute program under different environments.

Haswell

$\cdot 10^5$

Cycles

8

7.5

7

0 1,024 2,048 3,072 4,096 5,120 6,144 7,168 8,192

Bytes added to environment

SDC 17

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Bias from Environment Size III

- How to find the cause?
  - Accurate measurements
    - Hardware performance counters
  - Use correlation
    - Check *lots* of different CPU performance events
    - Correlate them with performance

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Bias from Environment Size IV

Performance counter	Median	Spike 1	Spike 2
ld blocks partial.address alias	137	327,871	327,848
cpu clk unhalted.thread p	692,686	825,311	823,872
uops executed port.port 0	240,248	106,477	106,409
uops executed port.port 1	249,016	100,913	101,250
uops executed port.port 2	336,768	345,932	346,328
uops executed port.port 3	358,848	356,919	357,723
uops executed port.port 4	280,766	276,036	276,199
uops executed port.port 5	251,631	121,517	121,571
uops executed port.port 6	234,337	178,586	178,646
uops executed port.port 7	137,428	130,325	130,542
resource stalls.any	274,373	406,364	406,256
resource stalls.rs	272 371	135,567	136,503
cycle activity.cycles ldm pending	590 879	718,973	721,580
cycle activity.stalls l2 pending	465,205	490,004	491,656

SDC 17

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Bias from Environment Size IV

Performance counter	Median	Spike 1	Spike 2
ld blocks partial.address alias	137	327,871	327,848
cpu clk unhalted.thread p	692,686	825,311	823,872
uops executed port.port 0	240,248	106,477	106,409
uops executed port.port 1	249,016	100,913	101,250
uops executed port.port 2	336,768	345,932	346,328
uops executed port.port 3	358,848	356,919	357,723
uops executed port.port 4	280,766	276,036	276,199
uops executed port.port 5	251,631	121,517	121,571
uops executed port.port 6	234,337	178,586	178,646
uops executed port.port 7	137,428	130,325	130,542
resource stalls.any	274,373	406,364	406,256
resource stalls.rs	272 371	135,567	136,503
cycle activity.cycles ldm pending	590 879	718,973	721,580
cycle activity.stalls l2 pending	465,205	490,004	491,656

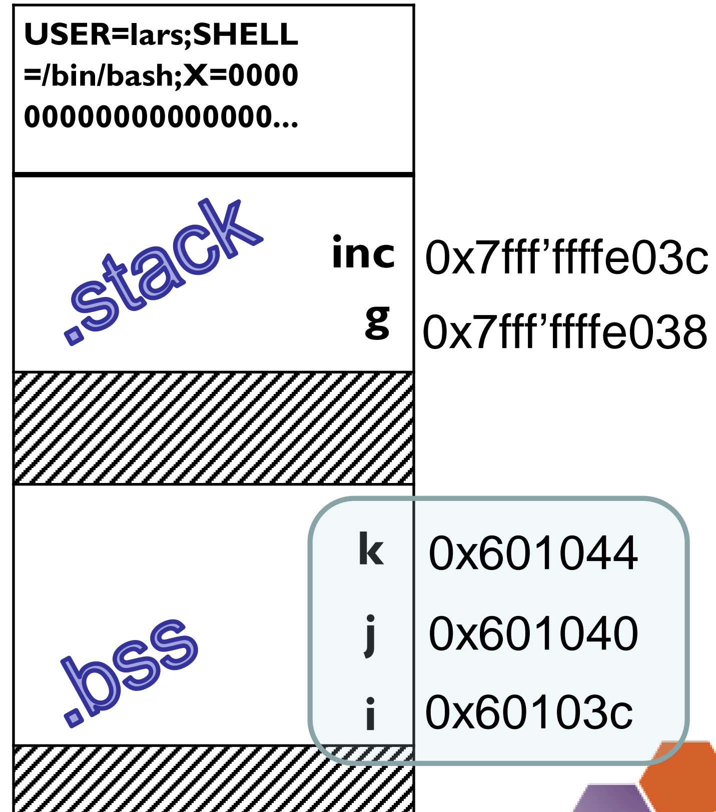
Correlation!

SDC 17

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

The cause: 4K Address Aliasing I

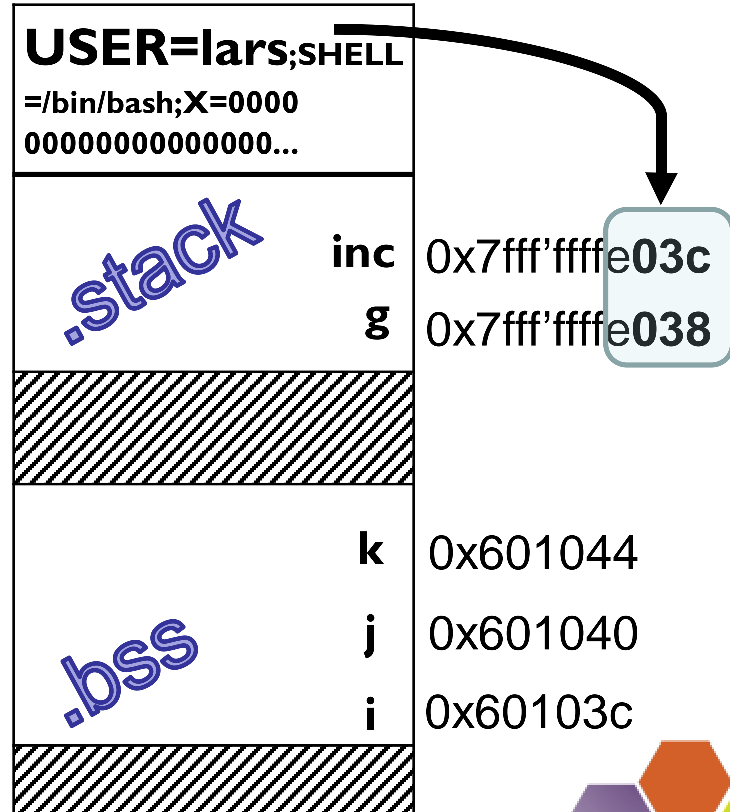
- Addresses of static ***i***, ***j*** and ***k*** are fixed for all environments.



2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

The cause: 4K Address Aliasing II

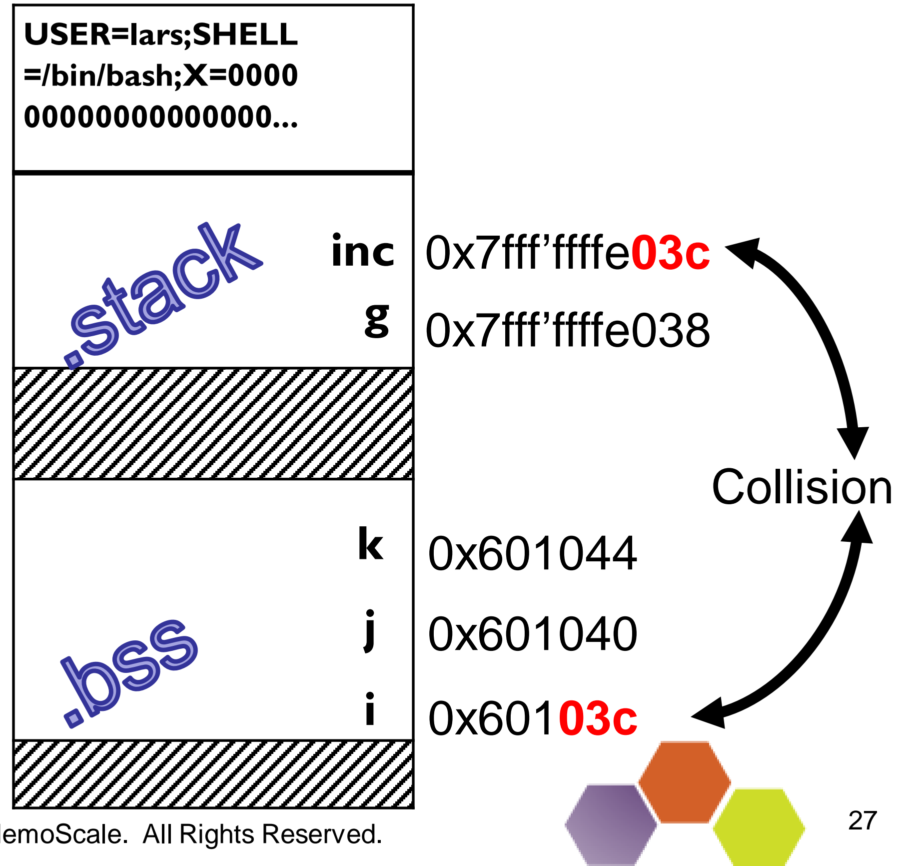
- Addresses of ***inc*** and ***g*** live on the stack.
- The stack is pushed down by the environment variables!



2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

The cause: 4K Address Aliasing III

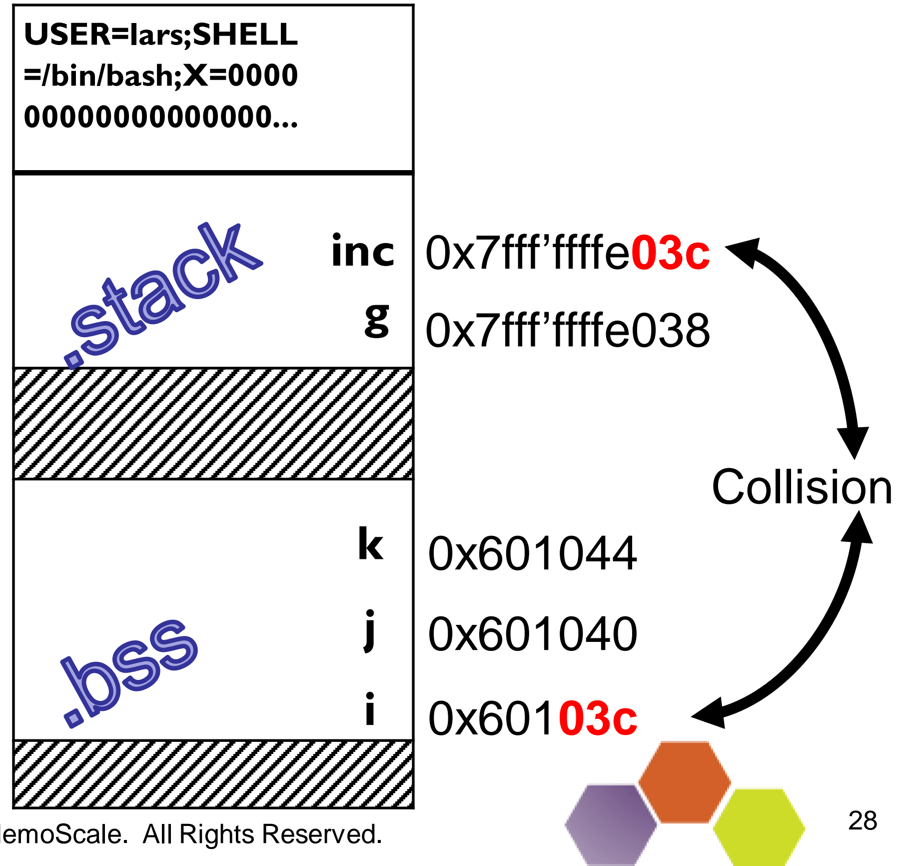
- Spike when address of ***inc*** is aliasing with ***i*** in the last 12 bits.
  - The last 3 hex digits.



2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

The cause: 4K Address Aliasing IV

- Performance impact from *false* dependencies in CPU.
- Hurts Out-of-Order execution.



2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

How to Measure Address Aliasing

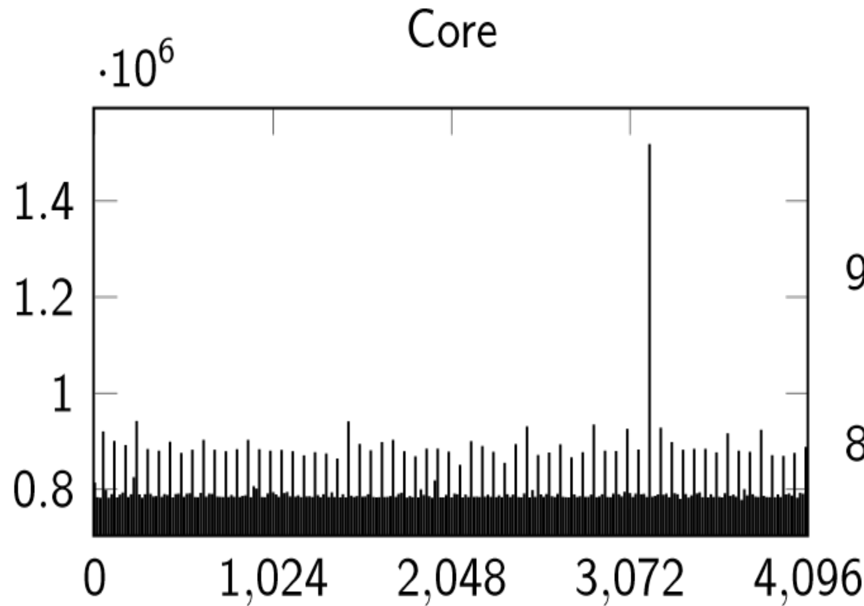
- Hardware counter event
  - Linux: perf
    - Manually add event type

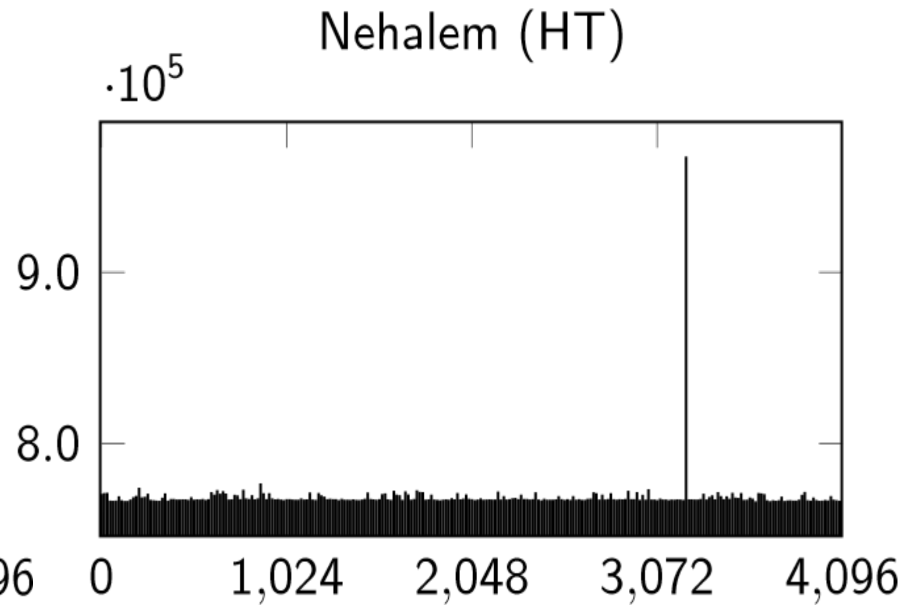
LD BLOCKS PARTIAL.ADDRESS_ALIAS

"Counts the number of loads that have partial address match with preceding stores, causing the load to be reissued." [3, B.3.4.4]

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Other architectures I





2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Other architectures II

Ivy Bridge

$\cdot 10^5$

y-axis	8.0, 7.5, 7.0
x-axis	0, 1,024, 2,048, 3,072, 4,096

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Bias from Heap Allocation I

- Alias conflicts can happen between any sections of memory.
- While stack addresses depend on environment, heap addresses depend on allocators.

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Bias from Heap Allocation II

- Addresses returned by different heap allocators when allocating pairs of equally sized buffers.
- Same 12 bit suffix indicate an aliasing pair.

	5,120 B	1,048,576 B
glibc (ptmalloc)	0x602010	0x2aaaaaaf6010
	0x603420	0x2aaaab096010
tcmalloc	0xe0e000	0xe0e000
	0xe10800	0xf0e000
jemalloc	0x2aaaabc0e000	0x2aaaabc06000
	0x2aaaabc10000	0x2aaaabd06000
Hoard	0x2aaaaab00070	0x2aaaaab00070
	0x2aaaaab02070	0x2aaaabf40070

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Bias from Heap Allocation III

- Most allocators tend to use anonymous memory mapping for large requests.
  - This performed by mmap.
  - Always at a page boundary.
- Since page size is 4 KiB, such allocations will *always* alias with each other.

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Bias from Heap Allocation IV

- Other reasons for 4K alignment
  - RDMA
    - Requires page locked memory
      - Naturally aligned to 4K
  - Drives use 4K blocks (not the same, but…)
  - Memory mapped files
    - mmap – always page aligned

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Optimization Resistant Code I

- First example program was too simple?
- Simplified implementation of convolution with a fixed kernel?

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Optimization Resistant Code II

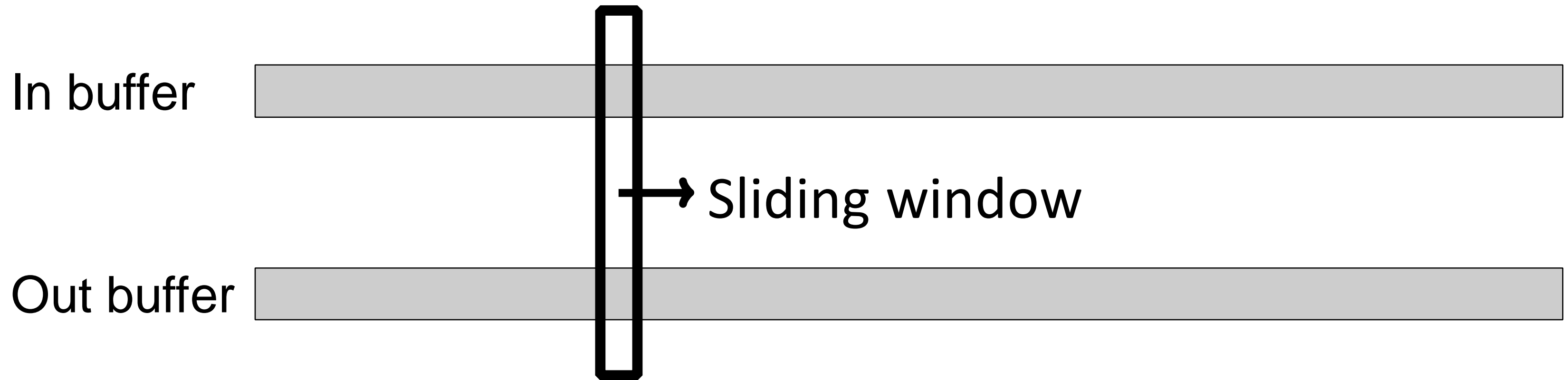
```
static float k[5] = {0.1, 0.25, 0.3, 0.25, 0.1};

void conv(int n, const float *in, float
   *out) { int i, j;
   for (i = 2; i < n - 2; ++i) { out[i]
      = 0; for (j = 0; j < 5; ++j) {
         out[i] += in[i-2+j] * k[j]; }
   }
}
```

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Optimization Resistant Code III

- The key property is the data access pattern:



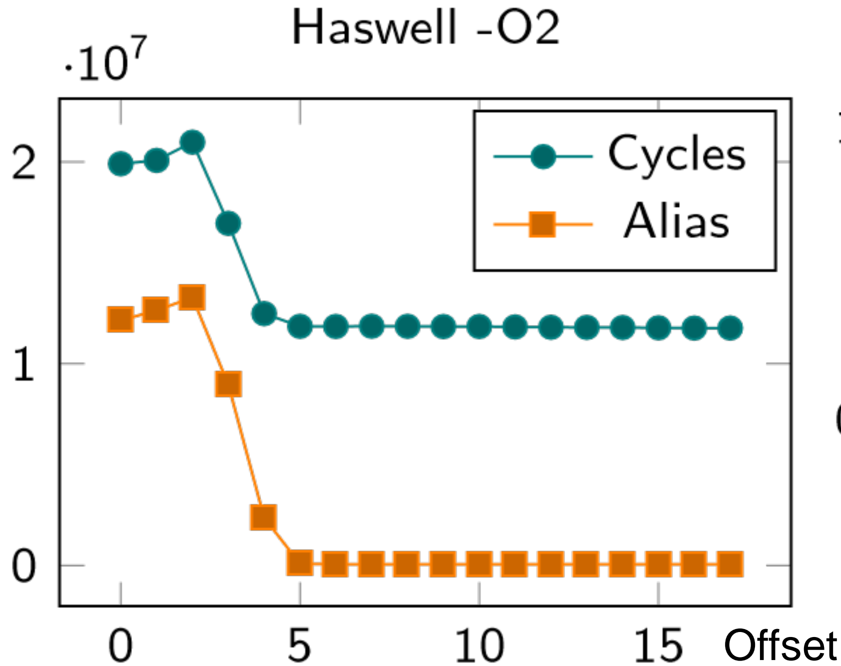
2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

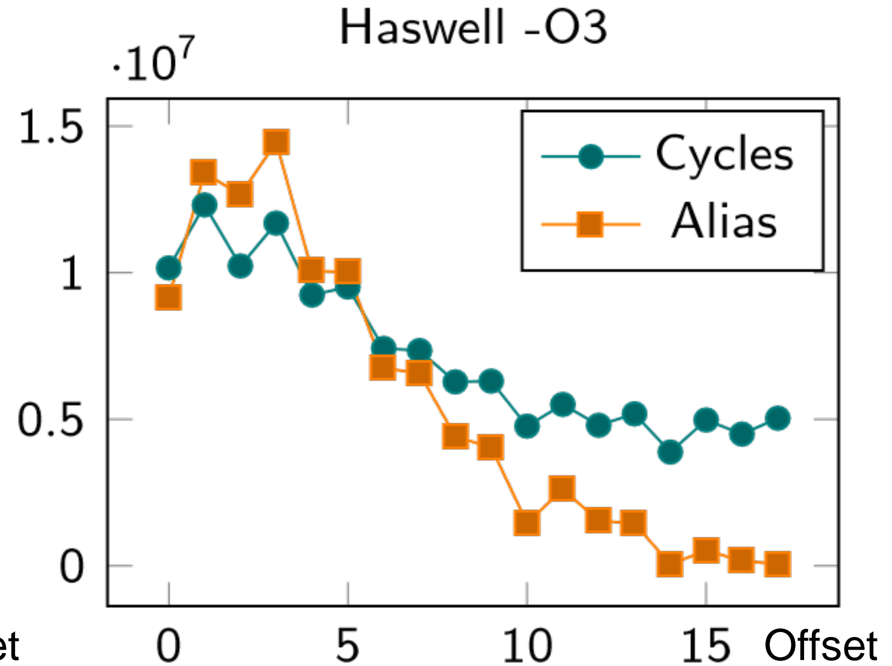
Optimization Resistant Code IV

- Adjust address of output array manually.
  - Offset arrays by *d* elements.
    - conv(N, in, out + d);
    - N = 2^20 (4 MiB data)
  - Evaluate performance.

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Optimization Resistant Code V





2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

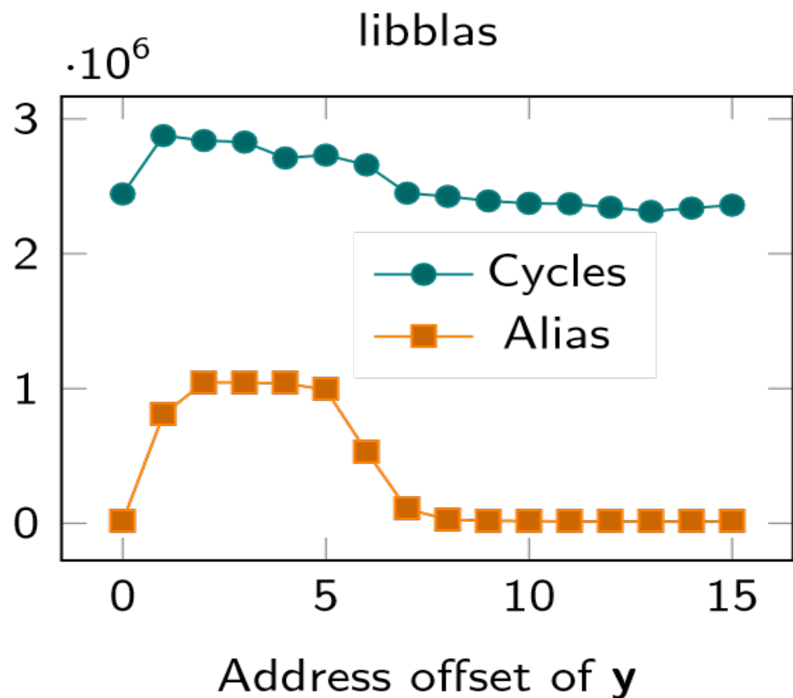
Aliasing in BLAS* libraries I

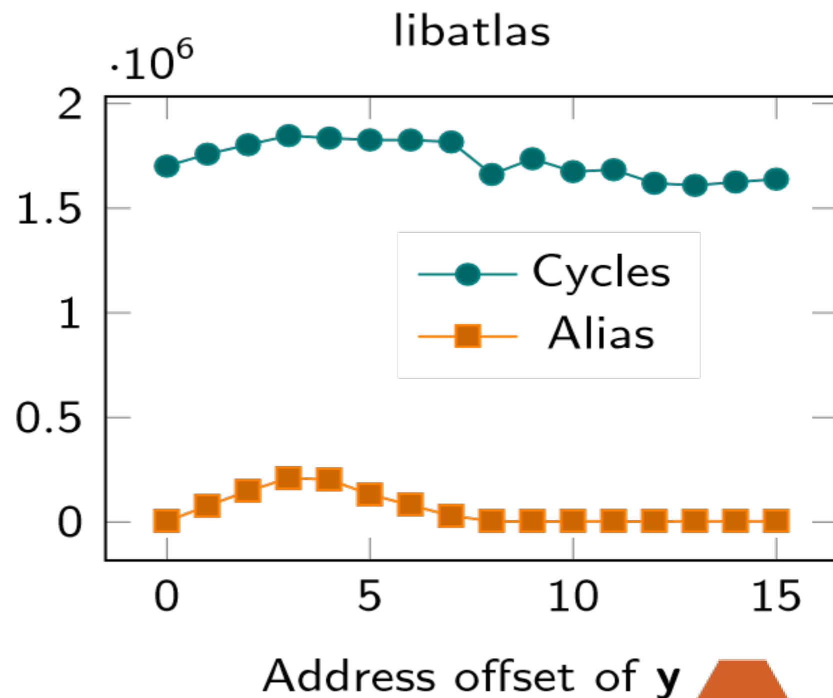
- What about heavily optimized libraries?
  - Measured matrix-vector multiplication (cblas dgemv) with increasing offset between buffer addresses. Computing **y** = **Ax**.

*Basic Linear Algebra Subprograms

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Aliasing in BLAS libraries II





2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Optimizing for address aliasing I

- Mark buffers with **restrict**
  - Can allow the optimizer to reduce number of memory accesses.
- Use optimized libraries
  - They might mitigate the effect.

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Optimizing for address aliasing II

- Use a special purpose allocator.
  - Heuristics for avoiding pairwise aliasing buffers.
  - No known implementation with this goal in mind, to our knowledge ☹

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Optimizing for address aliasing III

- Explicitly adjust address offset.
  - Consider memory address alignment for performance tuning inner loops.
  - Buffer addresses as candidate for automatic performance tuning.

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Learning points I

- Accurate benchmarking is hard.
  - Many uncontrolled effects creates bias.
  - CPU's are very complex.
- Optimizing is hard.
  - Too many rules to address at the same time.
  - Performance is a sum of good and bad cases.

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Learning points II

- We focused on address aliasing.
  - And some of the implications it had.
- There are more (and bigger) issues like this.
  - CPU cache!
  - Multithread locks ☹
  - TLB (Translation lookaside buffer).

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Learning points III / Summary

- Accurate benchmarks are crucial.
  - Large set of benchmarks.
  - Many measurement metrics needed.
    - Not only timings.
  - Identify performance issues (correlation).
    - Address aliasing (in our case).

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

References

1. T. Mytkowicz, P. F. Sweeney, M. Hauswirth, and A. Diwan, “Observer effect and measurement bias in performance analysis,” University of Colorado, Tech. Rep. CU-CS-1042-08, Jun. 2008.
2. T. Mytkowicz, A. Diwan, M. Hauswirth, and P. F. Sweeney, “Producing wrong data without doing anything obviously wrong!” in *Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems, ser. ASPLOS XIV.* New York, NY, USA: ACM, 2009, pp. 265–276.
3. Intel © 64 and IA-32 Architectures Optimization Reference Manual, Intel Corporation, Mar. 2014.
4. L. K. Melhus and R. E. Jensen, "Measurement Bias from Address Aliasing,“ *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, Chicago, IL, 2016, pp. 1506-1515.

SDC 17

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.

Q&A

- Further Questions…?
  - rune.jensen@memoscale.com
  - per.simonsen@memoscale.com

2017 Storage Developer Conference. © MemoScale. All Rights Reserved.